



Information Systems Laboratories, Inc.

RELAP5-3D Model Improvement

2012 International RELAP5 Users Group Meeting

Sun Valley, Idaho

October 23-24, 2012

Doug Barber

Information Systems Laboratories, Inc.

- Nodal Kinetics Improvements
 - Nodal Kinetics Data Restructuring
 - Purpose: Eliminate 1D container array (kn) for kinetics
 - Asynchronous Time Advancement
 - Purpose: Allow T/H and Neutronics to advance with different timestep sizes
- Wall Condensation Improvements
- Dissolved Gas Model
- New Command Line Options

Nodal Kinetics Restructuring

- Purpose: Eliminate 1D container array (kn) for kinetics
 - Arrays previously equivalenced to “kn” were moved to separate data structures
 - Improve readability, maintainability and robustness
 - Allow use of F95 run-time options (e.g. check bounds)
- Status:
 - Completed and delivered
 - Implemented in latest version of RELAP5-3D

Nodal Kinetics Restructuring

- Approach:
 - Initially, derived type structures with pointers were utilized to better organize the arrays according to functionality
 - This approach however introduced a significant CPU cost
 - Derived types were replaced by allocatable arrays
 - New allocatable arrays for each NK variable
 - In most cases variable names remained consistent with original pointer names
 - There were conflicts in a few cases that had to be addressed
 - New module k3dmod.F stores all the array data

Nodal Kinetics Restructuring

- k3dmod.F
 - Stores all NK arrays
 - Allocates / deallocates arrays
 - Reads / Writes data to restart file
- kinmod.F
 - “kn” array removed, along with all references
- k3parmod and k3ptmod.F no longer needed
- k3pntr.F was moved to k3dim.F
 - Pointer logic removed
 - Initializing of dimension information was retained
- irequest.F
 - Pointer references to “kn” array were replaced with pointers to the allocatable arrays
- Pointer reference mods also made to:
 - irkin.F, rkino.F, rkin.F, rrkino.F

Nodal Kinetics Restructuring

- Nearly every NK subroutine was impacted
 - Impact was mostly limited to removing variables from subroutine call list and removing local array declarations
 - Variable usage in the subroutines was largely untouched

abi1d.F, abi1dh.F, adjoint.F, ahnode.F, axb.F, axbh.F, axoffset.F, bcf.F, bicgstab.F, cheby1.F, chebytr.F, cmfdcoef.F, cmfdcofh.F, conver.F, d2tr.F, decusp.F, dir16.F, dir2full.F, dir4full.F, direct2x2.F, direct4.F, direct8.F, direct8b.F, dlaydn.F, errorn.F, fac1dh.F, factor1d.F, fbrhoc.F, fbrhoh.F, figout.F, gascac.F, gascath.F, gdhigh.F, gdmx.F, gdsun.F, geometry.F, geomkin.F, getfdbk.F, getfdbkp.F, getflux.F, getfsrc.F, givds.F, iluf2dh.F, ilufac2d.F, inital.F, initflux.F, initsrc.F, inittran.F, ipedget.F, k3dss.F, k3dtr.F, k3init.F, k3initvar.F, k3inputd.F, kgmres.F, kindex.F, krlvsol.F, lsorb.F, lsorb0.F, matinv.F, minv.F, minvh.F, nnkmout.F, nonnem.F, nonnemc.F, nonnemh.F, nonnemkh.F, nonnet.F, nonnetc.F, nonneth.F, nonnetkh.F, nonone.F, nononec.F, nononeh.F, nononekh.F, nonplm.F, nonplmc.F, nonplmh.F, nonplmkh.F, nontwo.F, nontwoc.F, nontwoh.F, nontwokh.F, norm.F, normfsp.F, outer.F, outerss.F, outin.F, outintr.F, output.F, outputss.F, outputtr.F, parcoss.F, parcstr.F, partran.F, pedata.F, pol1a.F, pol2a.F, polyxs.F, precr.F, precurs.F, putflux.F, putfsrc.F, rbmkuser.F, rbmkxs.F, react.F, reactfb.F, reduce.F, relpower.F, rfgout.F, scalp.F, setlstr.F, setlstr0.F, setlstrh.F, setup0.F, sol1d.F, sol1dh.F, sol2d.F, sol2dh.F, sorce.F, sorce0.F, storeic.F, sxenon.F, tpenbc.F, tpendriver.F, tpeninit.F, tpenonenode.F, tpenpflx.F, tpenresid.F, tpenwp.F, tpenupddhat.F, tridia.F, tridia0.F, txenon.F, update.F, upddiag.F, updfsrc.F, userxs.F, wiel.F, xsection.F, xsfdbk.F

- Changes were made on a piece-wise basis related to functionality (i.e., geometry, cross sections, linear system, nodal, etc.)

Nodal Kinetics Restructuring

- Verification Testing
 - Regression analysis on code versions before and after restructuring
 - Comparison of statistical parameters
 - Comparison of output files
 - 118 nodal kinetics test cases
 - Cartesian & hexagonal geometries
 - 2- & 4-group cases
 - Krylov and LSOR CMFD solvers
 - NEM & TPEN nodal solvers
 - BiCGStab & GMRES linear system solvers
 - With and without control rod decussing
 - Restart cases

Nodal Kinetics Restructuring

- Results of Regression Testing
 - Statistical Parameters
 - 20 parameters examined, including reactor power and core k-eff
 - No differences in any of the parameters except CPU time
 - Output File Comparison
 - Zero differences
- CPU Timing
 - Threshold: greater than 10 sec with +/- 5% difference
 - 10 cases with a CPU improvement
 - max: -24.3% (neacrp-c1-krlv-nem.i)
 - 22 cases with a CPU degradation
 - max: +24.0% (vver440-tr-hzp-krlv-cmfd.i)



Asynchronous Time Advancement

- Purpose: Allow T/H and Neutronics to advance with different timestep sizes
 - Allow T/H to subcycle Neutronics
 - Allow Neutronics to subcycle T/H
 - Synchronization required at regular intervals
- Status:
 - Completed and delivered
 - Implemented in latest version of RELAP5-3D

Asynchronous Time Advancement

- Functional Capability
 - Kinetics time step sizes are not required to be whole multiples or fractions of the thermal-hydraulics time step
 - $Dt_{th}=0.100s$; $Dt_{kin}=0.125s$ or $0.075s$
 - Steady-state or transient
 - Point or nodal kinetics
 - User can input absolute time step sizes or ratios
 - Absolute time step sizes used internally
 - Trip unit numbers for locking and unlocking time steps
 - Control variable for computing kinetics time step size
 - Synchronization points (user controlled and internal)
 - Exponential flux extrapolation (optional)
 - Special thermal-hydraulic feedback treatment
 - Detailed printing of time step control information

Asynchronous Time Advancement

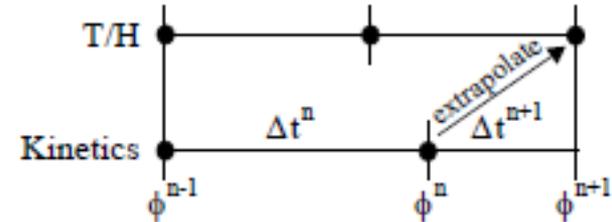
- Control Variable Usage
 - Provides greater flexibility and is likely the preferred modeling choice
 - **Warning**: there is an inherent lag which is magnified for the asynchronous time step control due to:
 - Kinetics time step size changes occur during the kinetics advancement
 - Control variables are updated at the end of a transient advancement
 - Kinetics time step size may be larger than T/H , thus skipping kinetics advancement
 - A return value of -1.0 returns control to the kinetics time step cards
 - A return value of 0.0 indicates the time steps are to be locked
 - Trip signals will override control variable usage

Asynchronous Time Advancement

- Synchronization points (user controlled and internal)
 - User controlled:
 - Input kinetics time step as a whole multiple or fraction of T/H
 - Use a control variable with programmed synchronization
 - Internally controlled:
 - End time on kinetics time step card
 - Major or Restart edits (not plot or minor edits)
- Integer time comparisons
 - Similar to logic used in `dtstep.F`
 - Improves precision of asynchronous control

Asynchronous Time Advancement

- Exponential extrapolation
 - Performed only when:
 - T/H advancement is ahead of the kinetics advancement
 - Next planned kinetics step will not occur until after T/H step
 - If multiple T/H per kinetics, extrapolation will be recomputed for each T/H step
 - Kinetics advances based on previous solution, not extrapolated solution
- Thermal-Hydraulic Feedback
 - Kinetics uses T/H feedback based on most recent T/H advancement prior to or at current kinetics time step
 - No extrapolation is performed



Asynchronous Time Advancement

- Cards 2200-2299, consistent with 200-299
 - Card 2200:
 - W1(I): Kinetics time step control option.
Currently, the only acceptable option for this word is 0, which indicates that manual kinetics time control logic will be utilized. In the future, a value of 1 could be used to specify automatic kinetics time step control logic.
 - W2(I): Kinetics time step format.
There are two options for this input. A value of 0 indicates that absolute time step sizes (s) will be entered on cards 2201-2299. A value of 1 indicates that a time step ratio equivalent to the target kinetics time step size divided by the target thermal-hydraulics time step size will be entered on cards 2201-2299. The default value for the time step format is 0 (absolute time step sizes).
 - W3(I): Trip unit for locking time steps.
This input specifies a trip unit number that locks the kinetics and thermal-hydraulics time steps when the trip is set, and reverts back to the 2201-2299 cards when the trip is not set. The default value for the locking trip unit is 0.
 - W4(I): Trip unit for unlocking time steps.
This input specifies a trip unit number that unlocks the kinetics and thermal-hydraulics time steps when the trip is set AND the time steps were previously locked by the trip from W3. The default value for the unlocking trip unit is 0.
The combination of W3 and W4 is useful for cases such as scram where the time steps need to be locked during a period of rapid change, and then can revert back to asynchronous advancement after the period of rapid change is complete. Note that if W3 is set to 0, this word has no meaning and should also be set to 0.

Asynchronous Time Advancement

- Card 2200 (cont' d):
 - W5(I): Control variable unit number returning time step size.

The return value from the control variable should always be absolute time step size (s), regardless of the value of W2. In addition, if a value of 0.0 is returned from the control variable, then the time steps will be locked, and if a value of -1.0 is returned, then the time step control from the 2201-2299 cards will be utilized. The default value for the control variable is 0.
 - W6(I): Flux extrapolation option.

There are currently two options for the flux extrapolation. If W6 is 0, then no extrapolation of the flux will be performed. If W6 is 1, then exponential flux extrapolation will be performed. The default for this option is 0.
 - W7(I): Time step data print flag

There are currently two options for this print flag. If W7 is 0, then the asynchronous time step data will not be printed. If W7 is 1, then the asynchronous time step data will be written to a file whose name is based on the input filename with extension *.dtk (if 'indta' is used for the input file name, then the time step data will be written to 'dtkdata'). The default value for this print option is 0.

Asynchronous Time Advancement

– Card 2201 - 2299:

- W1(R): Kinetics end time for this set (sec).

This number must be greater than or equal to zero. This quantity must increase with increasing card numbers, and the final end time value for the 2201-2299 cards must be the same as the final end time on the 201-299 cards.

- W2(R): Target kinetics time step size (sec) or ratio (-).

If W2 on Card 2200 is 0, then input absolute time step sizes (s) here. If W2 on Card 2200 is 1, then input a ratio equivalent to the target kinetics time step size divided by the target thermal-hydraulics time step size from W3 on Cards 201-299. If the absolute time step size or ratio entered here is set to 0.0, then the code will lock the kinetics and thermal-hydraulic time steps until the end time on this card set is reached.

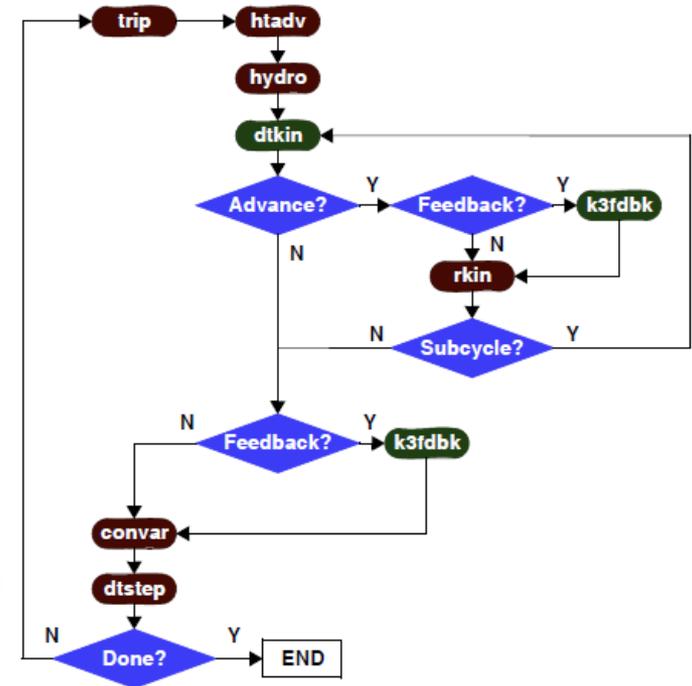
Asynchronous Time Advancement

- Module Source Modifications
 - ctrlmod.F: Added kinetics time step variables
 - idtmod.F90: Added kinetics integer time
 - k3dmod.F: Added two arrays for extrapolation
 - kinmod.F: Added kinetics advancement control variables
 - tscmod.F: Added kinetics time step card variables
 - ufilsmod.F: Added unit number of diagnostic data
- Kin3D Source Modifications
 - flxextrp.F: New routine for flux extrapolation
 - getcrod.F & getcrod.F: New routines split from getfdbk.F & getfdbkp.F
 - ik3crod.F & k3crod.F: New routines split from irkin and rkino, respectively for computing control rod positions
 - ik3fdbk.F & k3fdbk.F: New routines split from irkin and rkino, respectively for updating feedback parameters
 - powertr.F: New routine for updating power densities with logic extracted from rkino, k3dtr and parcstr.
 - kgmres.F, outer.F, polyxs.F & rbmkuser.F: Replaced timehy with timekn

Asynchronous Time Advancement

- RELAP Source Modifications

- rdtkin.F: New routine to process kinetics time step control cards
- idtkin.F: New routine to initialize time step control logic
- dtkin.F: New routine for controlling kinetics time step advancement
- irequest.F: Added 'countk', 'dtkin' and 'timekn' to minor edit / plot variables (vreqd.H & vreqs.H)
- irkin & rkino.F: Removed logic for computing control rod positions and update feedback parameters; Added call to powertr.F
- majout.F: Added kinetics advancement data to the major edit header
- rnewp.F: Zeroed out new kinetics time step variables
- tran.F: Added new logic for kinetics advancement
- trnctrl.F: Modified advancement message; close diagnostics file



Asynchronous Time Advancement

- Verification Testing
 - 41 test cases devised
 - Test asynchronous advancement logic (23 cases)
 - Test error checking logic (18 cases)
 - Logic tested with typpwrr2 base case
 - Time steps locked ($dt_{kin}=0.0$)
 - Asynchronous via time step cards (short & long cases)
 - Asynchronous via control variable (4 cases)
 - Flux extrapolation (short & long cases)
 - Time step ratios utilized
 - Kinetics subcycling T/H and vice versa
 - Trip logic (locking and unlocking)
 - Steady-state and transient restarts
 - All cases performed as expected



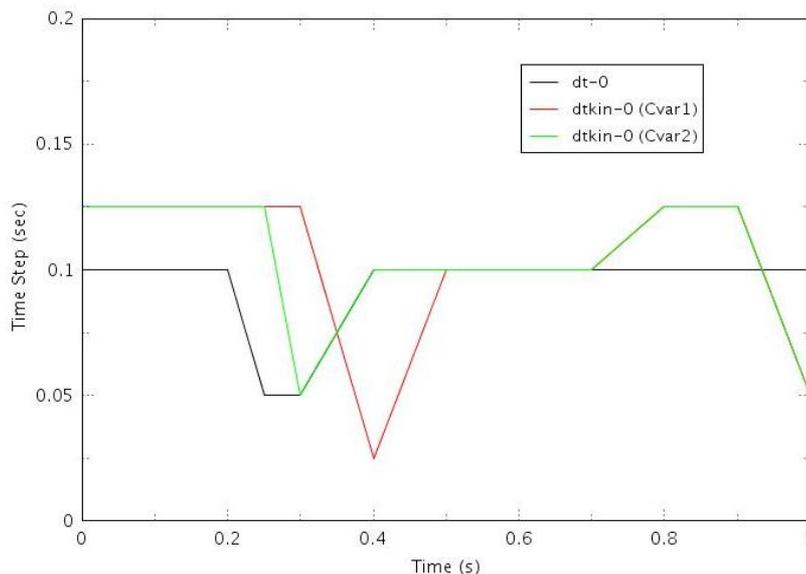
Asynchronous Time Advancement

Cvar1

*	time (sec)	kinetics time step size (sec)
20200301	-1.000	0.125
20200302	0.000	0.125
20200303	0.299	0.125
20200304	0.300	0.000
20200305	0.599	0.000
20200306	0.600	-1.000
20200307	1.000	-1.000
20200308	1.0+6	-1.000

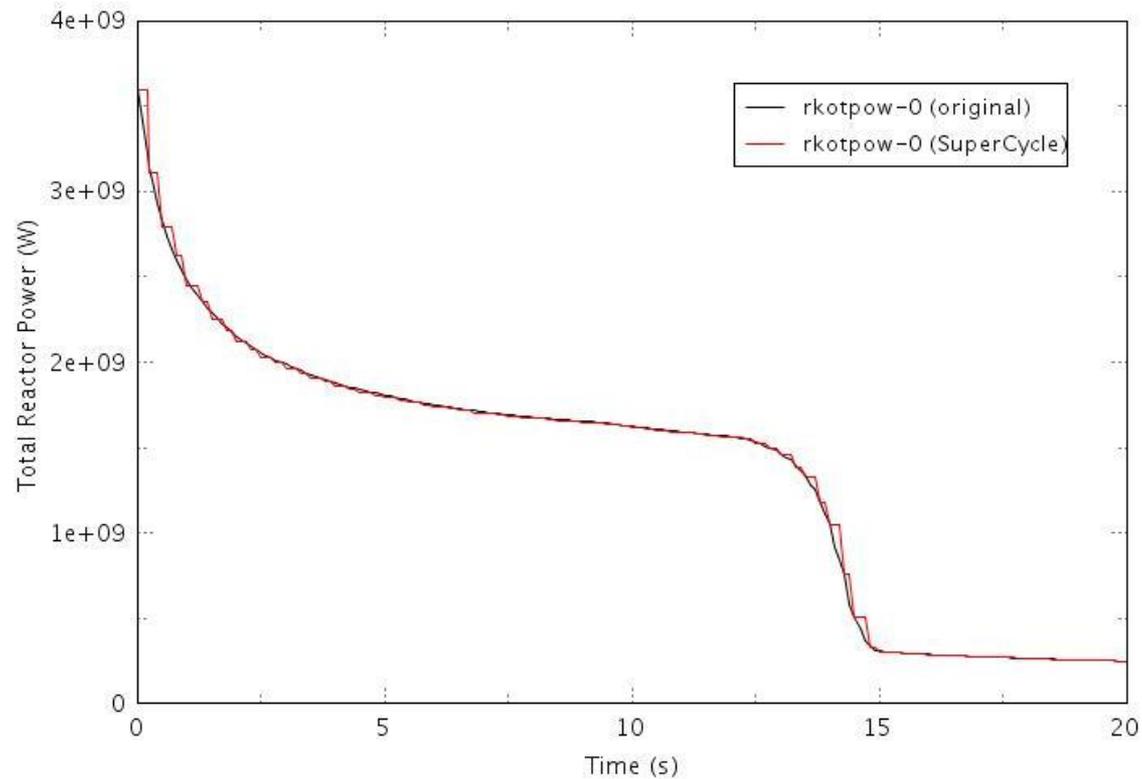
Cvar2

*	time (sec)	kinetics time step size (sec)
20200301	-1.000	0.125
20200302	0.000	0.125
20200303	0.249	0.125
20200304	0.250	0.000
20200305	0.599	0.000
20200306	0.600	-1.000
20200307	1.000	-1.000
20200308	1.0+6	-1.000



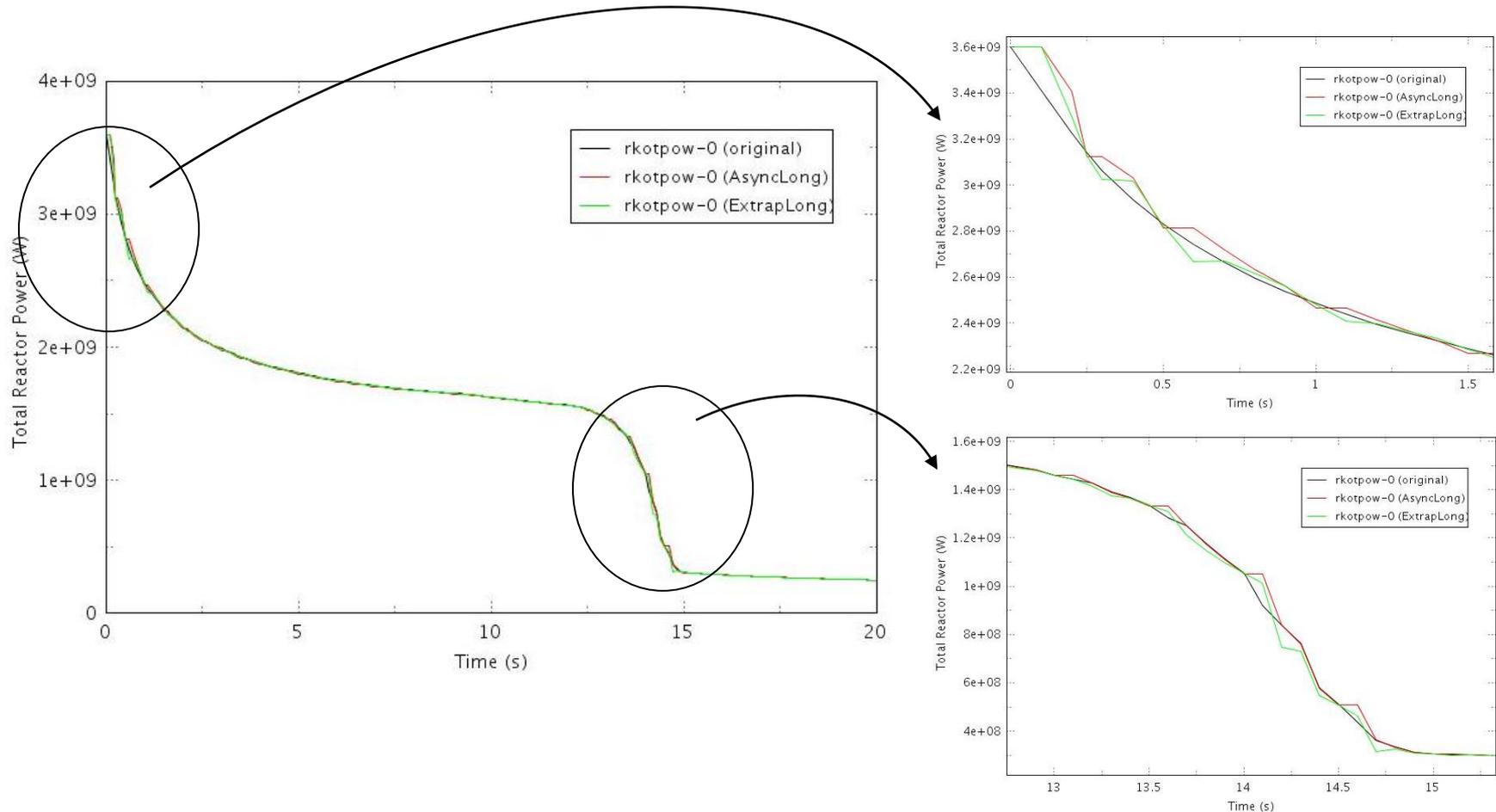
Asynchronous Time Advancement

- Kinetics Super-cycling T/H
 - T/H: 0.100s; K_{in} : 0.250s



Asynchronous Time Advancement

- Asynchronous with and without Extrapolation



Asynchronous Time Advancement

- Regression Testing
 - Same set as for NK restructuring (118 nodal kinetics test cases)
 - Statistical Parameters
 - 20 parameters examined, including reactor power and core k-eff
 - No differences in any of the parameters except CPU time
 - Output File Comparison
 - Zero differences
- CPU Timing
 - 7 cases with a CPU improvement
 - 15 cases with a CPU degradation
 - Negligible differences



Asynchronous Time Advancement

- Future Consideration
 - Automatic time step control
 - Additional flux extrapolation options
 - Improve T/H feedback logic



Condensation Model Improvements

Lance Larsen
Information Systems Laboratories, Inc.

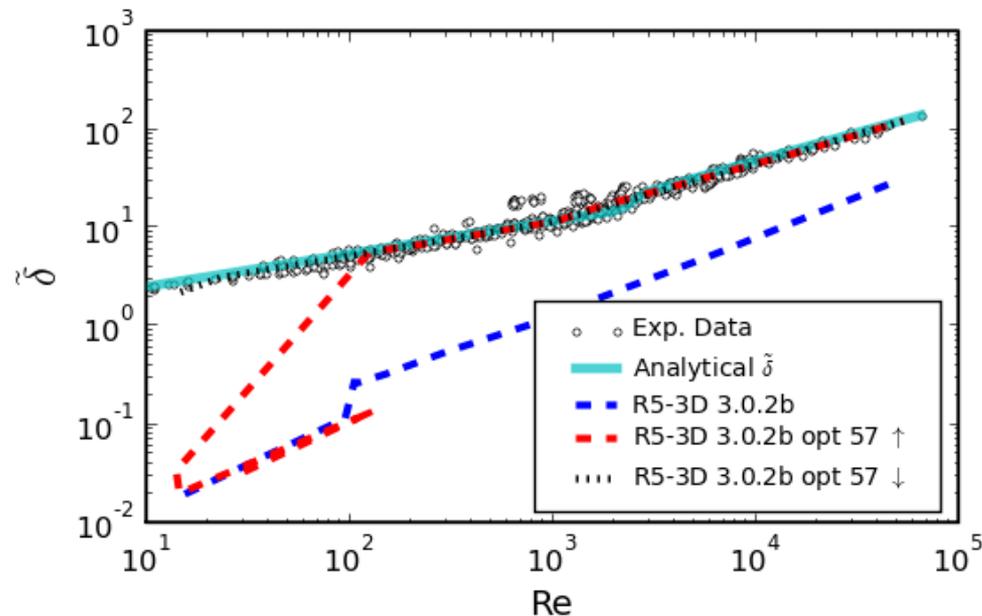
Condensation Model Improvements

The following condensation model improvements were implemented in order to improve agreement with the UCB-Kuhn condensation test data:

1. Card 1 option 75 was added, which causes the Kuhn-Shrock Peterson Nusselt correlation $Nu=1+1.83E10^{-4} Re_f$ to be used in place of $Nu = 1$ for condensation during laminar film flow.
2. Card 1 option 57 (the Wallis film flow friction factor correlation) was modified. The Wallis correlation gives much better friction factor predictions than the default HTFS model in the annular-mist regime. However, the Wallis correlation was disabled during countercurrent flow. The code was modified such that the Wallis correlation is active for both cocurrent and countercurrent flow in the annular-mist regime.
3. Because of poor film thickness predictions, the code used an analytical film thickness formula for the Nusselt condensation correlation. When card 1 option 57 is enabled, the code now uses predicted film thickness.
4. For thin film flow, $96/Re$ is used instead of $64/Re$ with interpolation used over void fractions from 99% to 95%.

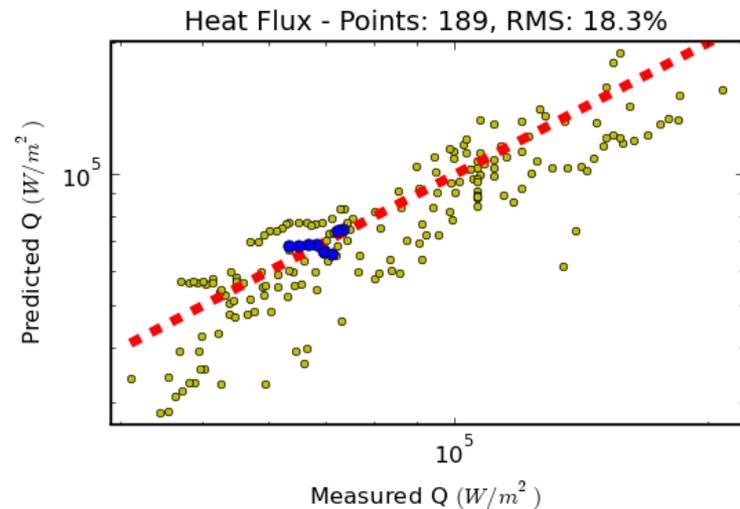
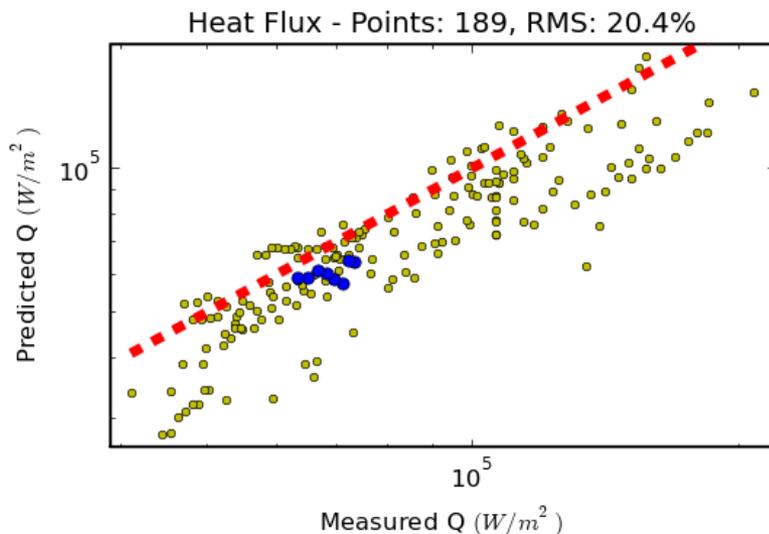
Film Thickness Results

Comparison of the default RELAP5-3D prediction (using the HTFS friction factor correlation) against the analytical solution, the film thickness data, and the updated card 1 option 57 solution is shown below. The Wallis correlation results are very good. The HTFS correlation RELAP5-3D results are poor (off by 1 to 2 orders of magnitude).



Baseline UCB-Kuhn Test Results

The UCB-Kuhn and MIT condensation tests were used to assess the condensation performance of RELAP5-3D. Data for 28 UCB-Kuhn test cases were available. The default RMS error for all cases combined is 20.4%. When card 1 options 57 and 75 are used, the RMS error is reduced to 18.3%. As an example, the blue dots represent the axial heat flux for UCB-Kuhn test 1.1-2. There is a clear improvement in predicted vs. measure heat flux for this test case.





Dissolved Gas Model

David Caraher
Information Systems Laboratories, Inc.



Dissolved Gas Model Added to RELAP5-3D

- Equilibrium concentration From Henry's Law
 - Concentration is dependent on partial pressure of the gas
- Model is incorporated implicitly in the semi-implicit numerical scheme
- Model can treat up to 5 dissolved gases -
 - Number of gases is user specified
 - Concentration continuity equation solved for each dissolved gas
 - Non-condensable vapor continuity equation for each dissolvable gas
 - Up to 9 new equations in the semi-implicit equation set
 - Number of extra NC_i continuity equations is one less than number of dissolved gases since X_a equation tracks sum of all NC_i

Dissolved Gas Model – Impact on R5/3D Coding

- Implicit Source term implies:
 - Additional terms (rows & columns) in the Jacobian (Preseq)
 - Additional derivatives needed
- Arbitrary Number of NCs implies:
 - Dynamic Jacobian Structure
 - Recursive LU Decomposition and Back Substitution
- Extensive Modifications made to Preseq & Eqfinl
 - Jacobian is now array a_(neqt_, neqt_, cell)
 - GuassElim subroutine replaced
- Modifications made to statep to compute Derivatives

$$\left(\frac{\partial T_g}{\partial X_{nj}} \right)_{P, U_g, X_n, X_{ni, i \neq j}}$$

$$\left(\frac{\partial \rho_s}{\partial X_{nj}} \right)_{P, U_g, X_n, X_{ni, i \neq j}}$$

$$\left(\frac{\partial \rho_n}{\partial X_{nj}} \right)_{P, U_g, X_n, X_{ni, i \neq j}}$$

- Input Processing:
 - Allow DG concentrations to be specified for all components
 - Allow user specification of rate constants for the dissolving/absolving process
- PVM:
 - PVM coding altered to accommodate Dissolved Gas Model

Dissolved Gas Model – Test Problem

- Closed Container, half full of liquid with air dissolved in the liquid (3300 ppm) and pure oxygen as the cover gas

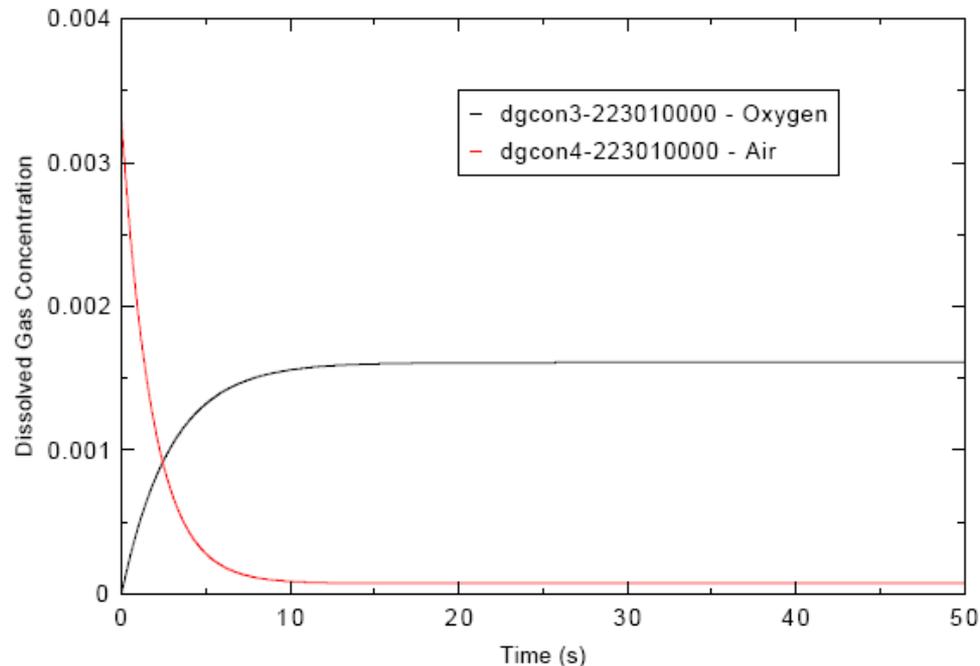
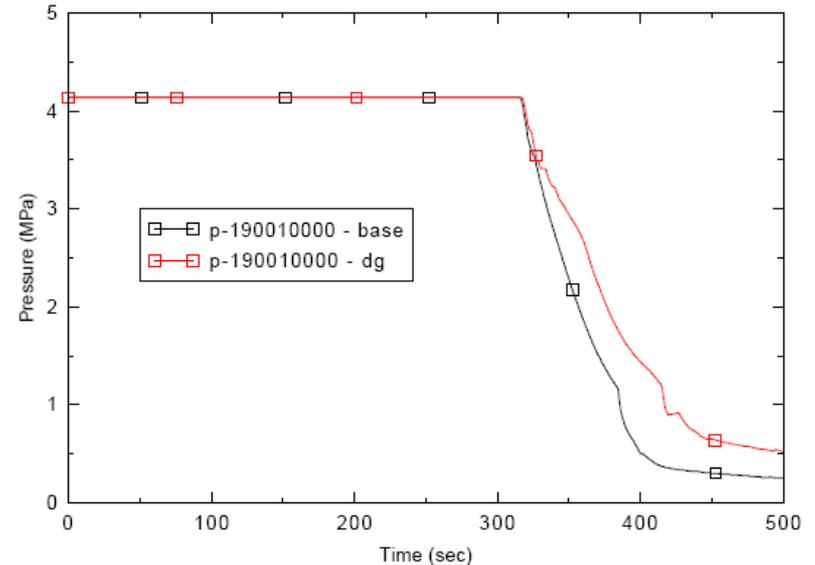
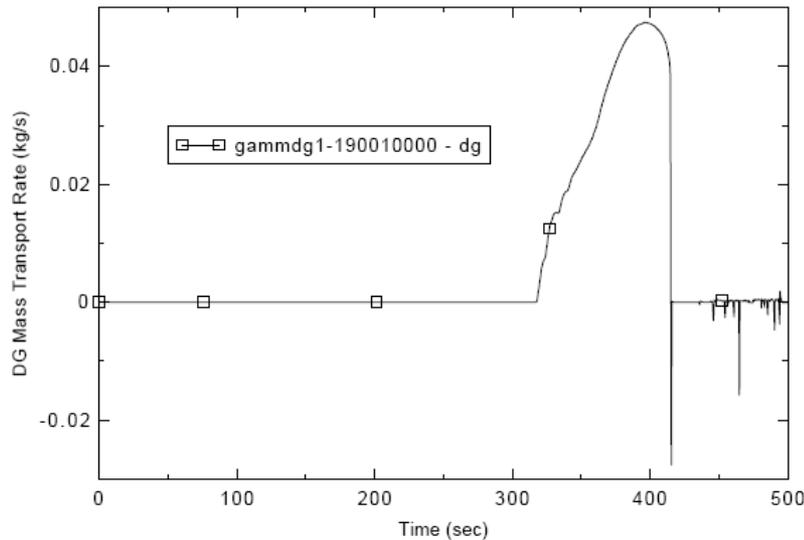


Figure shows oxygen going into solution and air coming out of solution
- Equilibrium in 10 s due to large rate transfer coefficients used

Dissolved Gas Model – TYP1200 Test Problem

- Dissolved gas desorption in accumulator during blowdown affects the rate of depressurization in the accumulator



Typ1200 test problem also indicated potential instability in the DGM when large, sudden changes in interfacial area occur due to flow regime changes



New Command Line Options

Lance Larsen
Information Systems Laboratories, Inc.

Command Line Help Message

Most command line programs have the ability to print a help message explaining the command line options. This capability was added to RELAP5-3D. To print the help message, any of the following may be used:

relap.exe --help

relap.exe -?

relap.exe -h

Sample of the Output:

RELAP5-3D command line options:

...

-t (default: tpflipb) File name for lithium-lead TPF

-u (default: tpfnh3) File name for ammonia TPF

-v (default: tpfvertrel) File name for vertrel TPF

...

-O Output file - overwrite existing output file

-P Neutron flux correction output file for RBMK

New Command Line Options

The following new command line options were added:

- **-rgtest**: This option causes date/time stamps, cpu time, and file name paths to be excluded from the output. This is useful for doing regression testing on models since it eliminates some irrelevant differences between output files.
- **-tpfdir [directory name]**: Directory where the tpf files are located. The path may be absolute or relative to the simulation working directory, or relative to the directory where the relap executable is located.
- **-O [output file name]**: An alternative to -o (lower case). When -O is used, the output file will be overwritten if it exists. With -o code exits if the output file currently exists.
- **-v [file name]**: (default: tpfvertrel) File name of the vertrel TPF.
- **-dtconf**: Path to the time step reduction reason log configuration file.